A Multi-Mission Space Avionics Architecture

Savio Chau, Kim R. Reh, Brian Cox
Jet Propulsion Laboratory

J.N. Barfield, W.L. Lockhart, M.L. McLelland
Southwest Research Institute

Abstract - As the budget for the space industry is dwindling, a reusable avionics architecture applicable across multiple missions is urgently needed to reduce the development and production costs of flight projects. This paper presents a multi-mission avionics architecture which employs interface standards extensively, so that avionics systems can be built rapidly by assembling subsystems and instruments in a plug-and-play manner. The key feature of this architecture is a "backbone" standard parallel bus which can accommodate various standard interfaces through a repertoire of 1/0 modules. Hence, specific mission requirements can be met by selecting and integrating the appropriate subsystems and instruments into the system. This architecture can also adopt different "backbone" buses, implement redundant processors, and support distributed processing. An example of the architecture using the MOPS6000 from the Southwest Research Institute is also described. Experiments will be done in the Flight System Testbed (FST) of the Jet Propulsion Laboratory to benchmark the MOPS6000 and to demonstrate its multi-mission capability.

Table of Contents

1. INTRODUCTION

As the budget for the space industry is dwindling, many new and exciting space programs do not have sufficient funding to develop new architectures and technologies to meet their requirements. Nor is there the market or funding to keep current suppliers interested in the qualification and production of electronic parts/components driven solely by space radiation effects. In order to circumvent the funding problem and, at the same time, develop the appropriate enabling technologies, JPI. has been working with industry to develop a reusable avionics architecture applicable across multiple missions. 'l'his architecture will be based heavily on the use of existing standards and will be designed to permit insertion of commercially available parts, where applicable. The architecture must also lean in the direction of tighter integration across classical subsystem boundaries. As a result, the associated development and design costs could be amortized and leveraged across these multiple missions and ideally across the larger commercial application market. This would result in lower costs for

individual missions, faster development cycles (e.g. twelve to eighteen months from preliminary design to acceptance test), rapid technology. insertion for upgrades and high quality/reliability avionics through inheritance and reuse,

The multi-mission concept is not new, but previous attempts at JPL have concentrated on identifying a single set of requirements (a point solution) suitable for multiple missions, as illustrated in Figure 1.

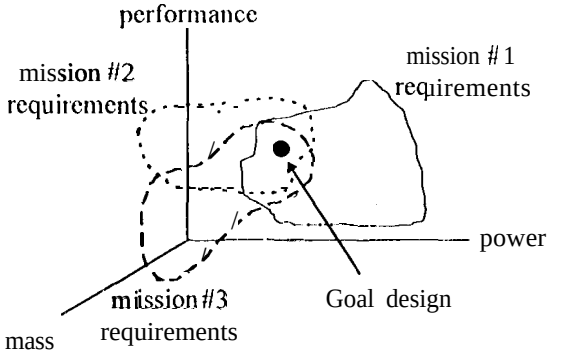

Figure 1: Traditional JPL Approach for Multiple Missions Avionics Design

Unfortunately the requirements for different missions rarely overlap, as it is shown in Figure 2.

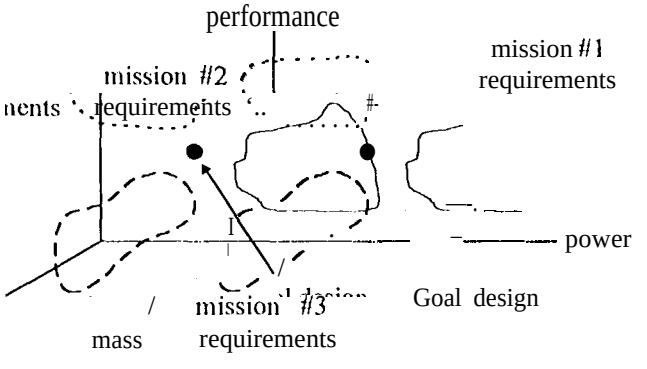

Figure 2: Mission Requirements Seldom Meet

In this paper, rather than trying to identify such a common and narrow set of requirements, wc propose to develop a flexible avionics architecture that can cover a much wider range of requirements in the trade space (a multi-dimensional region based solution). This is shown in Figure 3.

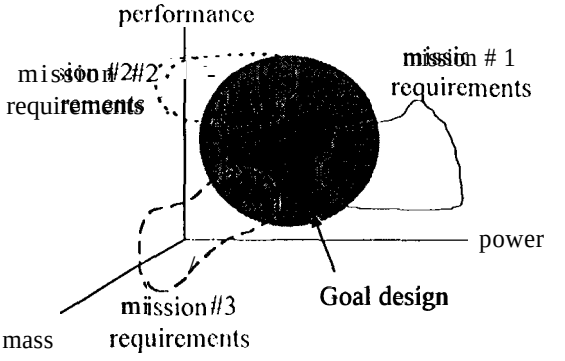

Figure 3: Flexible Core Design Covering Disjoint Mission Requirements

The proposed multi-mission space avionics architecture is being established by executing the following steps. First, establish the requirement space(s) that result from known planned space exploration missions or mission sets. Second, develop an architecture based on standards and as much existing avionics technology as possible that can fit within the boundary of these requirement spaces. Third, the communication infrastructure necessary to support the interface between subsystem element functions and instruments must be established, 'l'his infrastructure should be flexible enough to accommodate the rapid insertion of newly developed avionics clenlcnts/technologies. The interface between the avionics elements and the communication infrastructure should bc based on existing standards, if applicable, and must be specified as part of the architecturc. The resulting interfaces will then be capable of decoupling the avionics elements from the communication infrastructure, so that technologies within each clement can bc upgraded and inserted without causing a reinvention of the overall

core design of the system for each mission or mission set.

2. Survey of Spacecraft Avionics Requirements

A survey of the some of the avionics requirements for several planned missions at JPL is shown in '1'able 1.

These requirements include such items as mass, volume, and power of the information management system (i.e., Command and Data Handling (C&DH) and Guidance, Navigation, and Control (GN&C)), radiation tolerance, CPU performance, and the size of the local and mass memory. 'l'his survey is not exhaustive and many parameters have not been included. 1 lowever, it demonstrates the diversity of requirements for avionics systems in space missions.

3. A Multi-Mission Space Avionics Architecture

The summary of mission requirements shown in '1'able I suggests that a multi-mission avionics architecture must cover a large requirement space. In order to achieve a flexible architecture to cover disjoint requirement spaces imposed by different missions or mission sets, the system should be composed of reconfigurable modules and the interfaces of the modules must be standardized. These two requirements are the basis of the multi-mission space avionics architectures presented in this paper. An example of such a multi-mission architecture being considered by JPL is shown in Figure 4.

Table I: Survey of JPL Missions Requirements

	Cassini	Mars Pathfinder	Pluto Express	Champo-llion	Solar Probe	SIRTF
Information System Power (Watts/string)	38.0 (54)	13.75	5	3.3	3.25	5
information System Mass (Kg/string)	26.5 (32)	8.1	3	1.2	2	8
information System Volume (cm^3/string)	54406 (93081)	6600	< 2500	590	< 3000	8656
Radiation Tolerance (KRAD '1'11])	200	0.5	200	20	500	15
CPU Performance (MIPS)	1.2	10-22	> 5	> 5	< 4	> 2
Local Memory (MBytes)	1	128 MBytes	8	2.5	2	4
Mass Storage (Gbits)	2 (4)	(local+ mass)	4	0.128	4	1
Redundancy	dual string	single string	dual string	single string	dual string	dual string

Note: (1) Some of the mission requiremen arc preliminary and subject to change

(2) information system includes the Command and Data I landling function and the Guidance and Control functions

(3) Values represent one string of either single or dual string configurations. Values in parenthesis represent values for dual string configurations

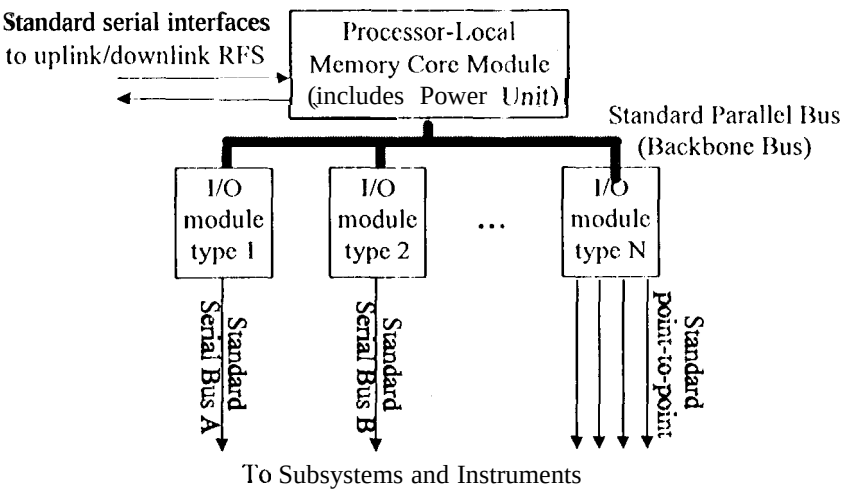

Figure 4: Multi-Mission Space Avionics Architecture

The architecture shown in Figure 4 is very similar to many space avionics architectures. The unique feature is that standard interfaces are used throughout the system. In particular, a standard parallel bus between the processor-memory and the 1/0 modules is used as the "backbone" bus which can accommodate a variety of subsystems and instruments. Therefore, subsystems and equipment with different mass, power, performance, and other characteristics can be integrated to meet the requirements of various missions without affecting the basic architecture of the avionics system. Hence, the avionics system can be built rapidly and cost-effectively.

The selection of a standard parallel bus as the backbone bus is an important architectural decision. The first standard parallel bus ever selected for flight projects at JPL is the VMEbus, which is used on the Mars Pathfinder project. Currently, the PCI bus is being considered by the first deep space mission of the New Millennium program, Pluto Express and other missions. On the other hand, it is also recognized that a true multi-mission architecture should be able to accommodate any new parallel bus standards. This can be achieved by employing a bridge between the old and new backbone standard parallel busses, as it is shown in Figure 5.

The extended multi-mission avionics architecture will allow a gradual migration from the old to the new backbone bus that is either more suitable for a particular mission or technologically advanced. An inverse approach can also be taken such that the new bus is the main backbone bus (i.e., directly controlled by the processor-memory core module) and a bridge is used to adopt the old bus so that previously designed 1/0 modules can be inherited. This approach is being taken by the first deep space mission of the New Millennium Program, in which the PCI bus is the new backbone bus and the VME is the old backbone bus.

Another factor that will affect the selection of the backbone bus is the number of masters on the bus. For missions that require distributed processing, the selected backbone bus must be able to support multiple bus masters. JJ)]., up to this time,

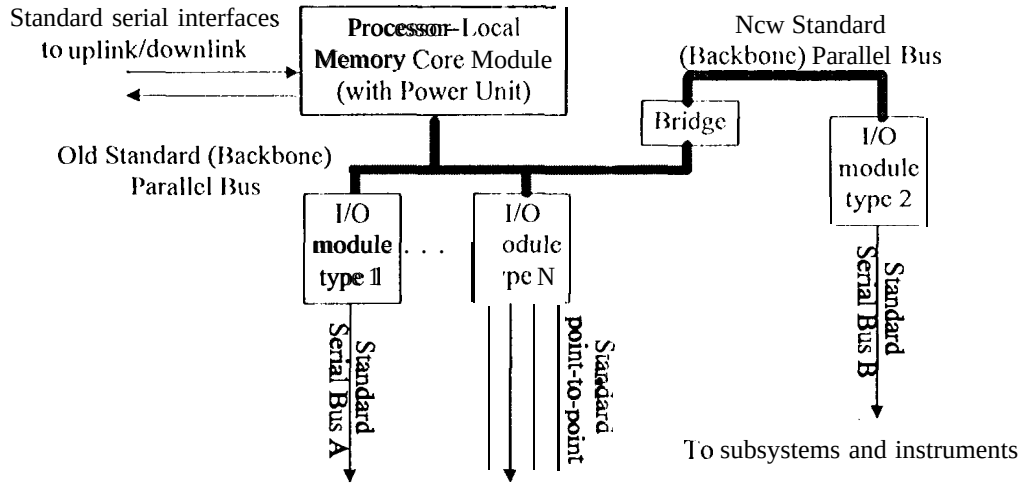

Figure S: Extended Multi-Mission Space Avionics Architecture

has not found any need for distributed processing in deep space missions. What most projects need is a centralized processor with software programmable speed and power. This is because in a typical deep space mission, the processor can run at very low speed during cruise phase to conserve power, and must run at full speed during the mission critical phase to capture the science and image data. As an example, the flight computer (MFC) being used in the Mars Pathfinder project has a programmable speed from 2.5 MIPS to 22 MIPS with an increment of 0.45 Watts/MIPS. The processor will run at 10 MHz during cruise and 20 MHz during the Entry-Descend-Landing phase.

On the other hand, as future spacecraft employ more autonomous techniques for navigation and fault protection to reduce the the need for ground intervention and hence the cost of mission operations, stringent requirements on computational capability will eventually require distributed processing. There are two possible ways for the multi-mission architecture to incorporate distributed processing capability. The first way is to introduce a new multi-master standard backbone bus to the system by means of a bridge as mentioned above. The other approach is to devise a special I/O module to facilitate the communication of the backbone buses of two or more processors. The latter approach is also suitable for implementing a dual-string fault tolerant avionics system (see Figure 6), which will be discussed in the next section.

Finally, the processor-memory core module must also have certain characteristics in order to be applicable for multiple missions. First, since mass and volume are always important concerns for any spacecraft design, the core module must take advantage of the most advanced technologies to minimize these parameters. Currently, there are several on-going projects at JPL planning to use the Multi-Chip-Module (MCM) technology to reduce mass and volume. Some of these projects are even planning to use MCM stacks to incorporate the processor-memory core and all I/O modules, so that the mass and volume can be further reduced. Some of the MCM stacks will adopt high density interconnects

and "Space-Cube" technologies. (Note: Space-Cube is a JPL invented technique in which the MCM modules can be rotated and stacked.) The selection of which MCM or MCM stacking technology will be based on an overall consideration of manufacturability, testability, reliability, and maintainability. Second, power consumption is another very important concern for deep space missions and therefore should also be minimized by advanced technologies. The 3.3 Volt technology is being planned to be used by several projects to reduce power. Third, the core module should have software programmable speed and power., These characteri sties arc not only needed to make the avionics system adaptable to different mission phases, they arc also needed to satisfy various performance and power requirements of different missions. Fourth, the core module should be designed in such a way that it can have either simplex or duplex processors. A core module with lock-step dual processors can simplify the system level fault tolerance while a single processor can reduce power consumption. These characteristics will have a wide range of applications and thus make the processor-mcmory core module applicable for multi-missions.

5. The Communications Infrastructure of Multi-Mission Avionics Architecture :

The purpose of establishing a communication infrastructure for the Multi-Mission Space Avionics Architecture is to provide the flexibility to support a wide range of interfaces for a variety of subsystems and instruments. This infrastructure includes the "backbone" parallel bus as described in the last section. in addition, it also includes the set of 1/0 modules and the software objects to drive these modules.

The necessary set of 1/0 modules can be very large if al 1 subsystems and instruments use unique and arbitrarily y designed interfaces. in order to limit the size of the 1/0 module set, it is necessary to restrict the designs of subsystems and instruments to standard interfaces. In this paper, it is assumed that the standard interface between the 1/0 module and the instruments are predominately serial, since the experience at JPL has shown that serial interfaces can significant y reduce the mass and complexity of cabling. 1 lowever, the backbone parallel bus does not exclude any instruments with parallel interfaces as long as the interfaces are widely accepted industrial standards.

With interface standards defining both ends of the 1/0 modules, a repertoire of pre-fabricated 1/0 modules can be established. Hence, avionics systems can be developed rapid 1 y by selecting and integrating these pre-fabricated modules with the process-memory core module. "l'his approach also provides some flexibility for the instruments to select the suitable interface standard. An example set of 1/0 modules that would be useful for some of the on-going and future projects at J] 'I. arc list' below. On the other hand, an industry wide effort will be needed to identify the set of most commonly used I/O modules in space applications and their form factors to facilitate the interchangeability.

(1) VME to 1553B bus
(2) PCI to RS422 (with UART)
(3) PCI to RS485 (with 155311 protocol)
(4) PCI to 1773 bus
(5) PCI to F ODB

The only interfaces that do not depend on the backbone parallel bus are the RFS uplink and downlink interfaces. They are separated from the backbone bus so that telecommand (uplink) and telemetry (downlink) will not be blocked by bus traffic. This is particularly important for the cases in which the spacecraft relies on ground intervention for fault recovery. There is no standard for these interfaces at this point. However, a typical uplink interface design used in many JPL deep space probes consists of data, clock, and synchronization signals. A typical downlink interface consists of data and clock signals. Since these interfaces are relatively simple, they could be easily adopted by most of the communication subsystems.

Corresponding to the repertoire of I/O modules, a library of software objects that control the 1/0 modules can also be established. The software objects will provide the basic services of the 1/0 module to the higher level applications. An 1/0 module can have more than one software object if it has several options of higher level protocols. Since the functionality of the 1/0 modules are well defined by the interface standards, the software objects are highly reusable. Thus, the software development cost can be significantly reduced. Such a software development effort is currently ongoing in the JPL Flight System Testbed.

Furthermore, the multi-mission avionics architecture has to support various levels of fault tolerance as required by many missions. This can be achieved by including a special cross-string 1/0 in the communication infrastructure. For instance, there are several projects at JPL employing dual-string architectures to achieve high reliability. The dual-string architecture has two strings of processor, local memory, mass memory, and various subsystems and instruments. The strings are redundant so that if one string fails, the surviving string becomes the active string. in order to support fault detection and recovery, both strings require a special I/O module that communicates with the other string. The application of the cross-string 1/0 modules in the multi-mission avionics architecture is depicted in Figure 6.

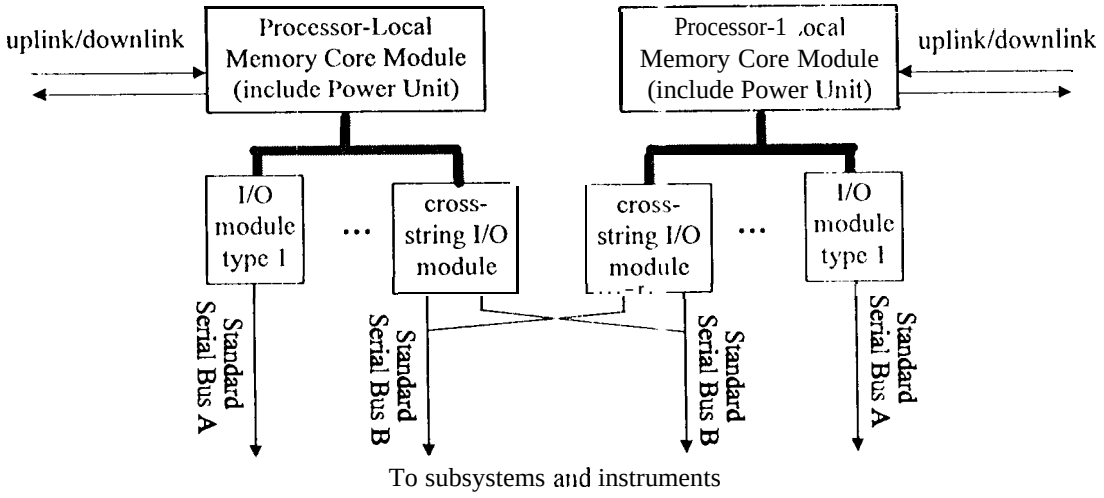

Figure 6: Dual-String Multi-Mission Space Avionics Architecture

6. EXAMPLE OF MULTI-MISSION AVIONICS ARCHITECTURE

As an instance of this multi-mission avionics architecture, Loral Federal Systems and the southwest Research Institute have developed the MOPS6000 (Miniaturized Optimized Processor for Space --RAD6000), and arc working with J]>], to further develop the MOPS6000 as a multi-mission space avionics system. Figure 7 shows the architecture of the MOPS6000, along with the block diagram of the LIO/VME bridge which will be explained in the following.

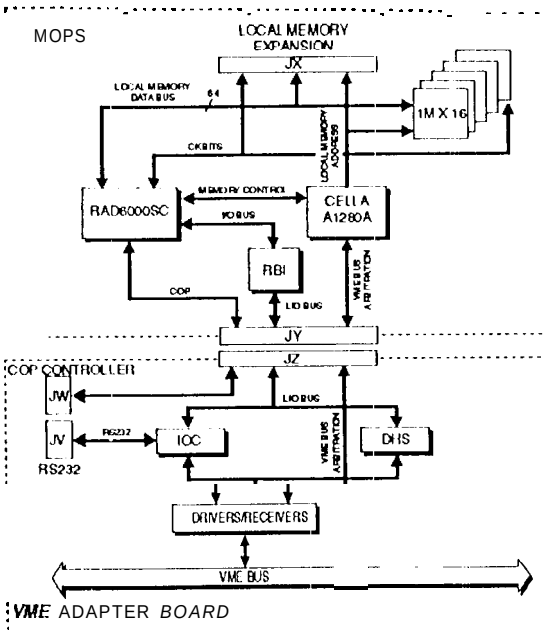

Figure 7: MOPS6000 Architecture Showing LIO/VME Bridge

The MOPS6000 is an ultra compact multi-chip-module-based space flight computer that uses the RAD6000 CPU engine developed by 1 oral Federal Systems. The MOPS 6 00 0 is less than 300 cubic centimeters (18 cu in.) in volume, with a mass < 350 grams. MOPS6000 delivers 2.0 MIPS pcr cubic inch and is functionally equivalent to the Mars Pathfinder Flight Computer which is currentl y mounted on a VMEbus 6U circuit board (9*7.7* 0.8 = 55.4 ci and 22 MIPS) and delivers 0.4 MIPS pcr cubic inch. Through this evolution of the avionics, the 6U size circuit board will event ual 1 y be replaced by an MCM stack and the VM Ebus will be replaced by the PCI bus. The salient features of the MOPS6000 are summarized in 'l'able II below.

The MOPS6000 communication infra-structure is made up two levels of interfaces. The first level is a high speed standard parallel bus for internal communication between the processor, local and mass memory, and the I/O modules. The MOPS internal parallel bus is the LIO bus. The MO bus will be bridged by FPGA to a PCI bus, or to one of the standard serial interfaces extending from the 1/0 modules to the instrument(s). Examples of these serial interfaces include MIL-STD-1 773A, UART with RS-422/485 for the electrical layer, or MI L-STD-1553B protocol with an 1{ S-485 electrical layer as a low power system. The 1/0 modules to handle these serial interfaces can be independently inserted into the PCI parallel bus, Hence, either point-to-point or bus topologies (or a hybrid of the two) can be implemented with this architecture. Since both the parallel bus and the serial interfaces arc existing standards, a repertoire of 1/0 modules for different standards can be pre-fabricated. 1 lence, the avionics system designer can select the 1/0 modules that can meet his own requirements such as data rate, power, and mass, and then put together the system rapidly in a "1 .EGO" approach. The protocol of the parallel bus is the interface between the communication infrastructure and avionics elements. As long as the processor, memory and 1/0 modules conform to this protocol, they can be upgraded without affecting the rest of the avionics system This plug-and-play approach may be further extended to include

Table 11: MOPS6000 Rad-Hard Spacecraft Computer

Heritage	Mars Pathfinder Flight Computer
Engine	RAD6000-SC
Memory	2 MBytes SRAM Main Memory 256 Kbytes EEPROM SUROM Memory Option: DRAM w/EDAC
I/O - Flexible/High PGA-based Interface	VMEbus, 1< S-232, 1< S-422/485, host interface, Internal Local 1/0 Bus
Speed	22 MIPS @ 20 MHz 35 MIPS @ 33 MHz
Software Selectable Clocks	1x, 1/2x, 1/4x, 1/8x
Power	10.5 Watts @ 20 MHz 2.5 Watts @ 2.5 MHz
Mass	< 350 grams
Size/Volume	~8 x 9 X 4 cm/300 cm^3
Software Development Tools Ada & C	C: VxWorks (Wind River) Ada: VADSCross or Equivalent RISC System 6000 or SUN Software Development Environment
Hardware Development Tools	RISC System/6000 Workstation. Mars Pathfinder Breadboard, Heurikon 68040 SBC

a multiple "cube" approach where similar packaging will implement other key functions of the system (i e., other Mars Pathfinder type functions/cards). This would allow a multiple cube approach on a "cold plate interconnect" with follow-on compression to an integrated or extended cube. A special 1/0 module is designed to allow direct communication between the parallel buses of two multi-mission avionics cores. This would allow implementation of multi-processor systems for distributed processing and/or redundant processors for fault tolerance. Much of the MOPS6000 design is based on a high level of inheritance from the Mars Pathfinder Attitude and Information Management Avionics. Flight computing ICS, ASICs, FPGAs, Flight Software, 1{"1'0S, development files/tools and integration and Test elements are being inherited where possible and reused or modified. As an example, this architecture provides the capability to reduce the mass of an AIM type system to 1/4 or 1/5 of the current MPF VMEbus approach while reducing the volume by a factor of at least two. It also results in an adaptable core that is easily reconfigurable for other missions and will provide the basis for a high percentage of inheritance,

The MOPS6000 prototype will be installed in the JPL Flight System Test Bed in the Fall of 1995 for extensive flight evaluation testing and bench marking. The first step of the experiment will including. porting the JPL FST software to the MOPS6000 on the testbed and performance of bench marking. The second step would include the insertion of a variety of 1/0 modules and proccssor-

memory core modules to run applications for various missions to demonstrate the multi-mission nature of the architecture. The last step will include experiments with multiple processors to demonstrate fault tolerance and distributed processing required to support cm-board autonomy functionality. The first step of the experiment is expected to bc done by early spring 1996. The other steps arc planned to be done in the following year. Results of the experiment will be published in near future.

7. Conclusion

This paper has presented a multi-mission avionics architecture which is derived from past experience, on-going flight projects at JPL and new mission studies. This architecture employs interface standards extensively. The key feature of this architecture is a backbone standard parallel bus which can accommodate various standard serial interfaces through a repertoire of I/O modules. The techniques to adopt new backbone bus, to implement redundant processors, and to support distributed processing with this multi-mission architecture have also been discussed. The flexibility of this architect ure will allow avionics systems to bc built to the requirements of most of the missions rapidly and economically. An instance of the architecture using the MOPS6000 from the Southwest Research Institute and Loral has also been described. Experiments to validate the multi-mission architecture concept using the MOPS6000 wi llbc performed in the JPL Flight System Testbed. Results of the experiment will be published in the near future.

Acknowledgment

The Southwest Research Institute has contributed a copy of the MOP S6000 for the multi-mission architecture experiment, Funding for the experiment is provided by the Flight System Testbed at JPL. Much of the material presented in this paper is obtained from several flight project design team leaders at JPL, which include James Donaldson from the Cassini, David Woerncr from the Mars Pathfinder, Douglass Caldwell from the New Millennium Progi am Architecture Development '1'cam, Shirley Peak from the Solar Probe, Humphrey Price from the Pluto Express, and Kim Aaron from the Rosetta/ Champollion.

References

[1] James Donaldson. "Cassini Orbiter Functional Requirement Book: Command And Data Subsystem", JPL internal dot. CAS-4-2006, June 28, 1994.

[2] "Mars Pathfinder Attitude and Information Management (AIM) Subsystem Detail Design Review", JPL internal document, June 28-29, 1994.

[3] "Pluto Express FY95 Report", JPL internal dot. D-1293 1, September 30, 1995.

[4] "Champollion: NASA/CNES Preliminary Proposal To ESA For A Surface Science Package For The Rosetta Mission", November 1, 1994.

[5] Doug Caldwell, "NMP Flight One Avionics Handbook, Version 1.1", JPL internal document.

[6] Gary Bolotin, "Space-Cube Architecturcs", JPL internal document (patent pending).

[7] Brian Cox, Savio Chau, Kim Rch, "Multi-Mission Avionics Architecture For The New Millennium Program", JPL internal document.